



Université Abou Bakr Belkaid– Tlemcen

Faculté des Sciences

Département d'Informatique

République Algérienne Démocratique et Populaire

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique Option :

Réseaux et Systèmes Distribués (R.S.D)

Thème

**Conception et exploitation d'un entrepôt de donnée
au sein de la banque BADR**

Réalisé par :

- **MAHMOUDI Nesrine**
- **BABA AHMED Farah**

Présenté le 14 décembre 2016 devant le jury composé de MM.

- | | |
|---------------------------|--------------------|
| ➤ BENMAMMAR Badr | <i>(Président)</i> |
| ➤ HALFAOUI Amel | <i>(Encadrant)</i> |
| ➤ EL YEBDRI Zeyneb | <i>(Examineur)</i> |
| ➤ MAATALLAH Hocine | <i>(Examineur)</i> |

Année universitaire : **2015-2016**

Résumé

L'entreposage de données est une phase du processus décisionnel qui supporte efficacement le processus OLAP, il est né dans les entreprises pour l'aide à la prise de décision. Ainsi, les principaux utilisateurs de ces technologies font partie intégrante de l'entreprise, Nous pouvons citer les secteurs qui peuvent tirer profit des outils d'aide à la décision comme les banques et les assurances, ainsi que ceux de l'automobile et des institutions médicales.

La réalisation de l'entrepôt de données au sein de la BADR BANK, facilitera la prise de décision, et donnera une vision plus claire sur la situation des crédits octroyés, en donnant la possibilité aux décideurs de **connaître** la rentabilité économique et financière de chaque agence donc la rentabilité par région, de savoir lequel des crédits est à financer le plus ou le moins.

ملخص

طبق في الشركات OLAP تخزين البيانات هو مرحلة من مراحل عملية اتخاذ القرار التي تدعم بشكل فعال عملية للمساعدة في اتخاذ القرارات. وهكذا، فإن المستخدمين الرئيسيين لهذه التقنيات هو جزء لا يتجزأ من هذا الأخير، يمكن أن نذكر القطاعات التي يمكن أن تستفيد من أدوات دعم القرار مثل البنوك والتأمينات السيارات و المؤسسات الطبية. تحقيق مستودع البيانات في بنك بدر، يسهل اتخاذ القرارات ويعطي صورة أوضح عن حالة القروض الممنوحة، بإعطاء الفرصة لصانعي القرار لمعرفة ربح كل وكالة بالتالي ربح كل منطقة، لمعرفة أي من القروض صالحة لتمويل أكثر أو أقل.

Abstract

Data warehousing is a phase of process decision-making that effectively supports the OLAP process was born in business for help decision-making . Thus, the main users of these technologies are an integral part of the latter. We can cite sectors that can benefit from decision-making tools such as banks and insurance, as well as those of the automobile and medical institutions. The realization of the data warehouse within BADR BANK will facilitate decision-making and give a clearer picture of the situation of the granted credits, giving decision-makers the opportunity to know the profitability of each agency thus the profitability by region, to know which of the credits is to be financed more or less.

Remerciements

Nos remerciements s'adressent à Dieux tout puissant qui nous a donné le courage, la force et les moyens pour réussir et donner le meilleur de nous-mêmes.

Nos remerciements s'adressent à notre encadrant Madame HALFAOUI Amel pour avoir accepté de diriger ce travail et pour ces compétences qui nous ont été d'une grande aide.

Nous tenons à remercier les membres du jury qui nous font l'honneur d'évaluer ce travail.

Ce travail n'aurait pas été possible sans le soutien de nos familles, c'est pour ça que nous tenons à remercier nos parents ainsi que nos proches pour tous les encouragements.

Nos remerciements les plus chaleureux vont à tous ceux qui nous ont apportés de l'aide de loin ou de prêt.

Et enfin nous tenons à remercier nos amis qui ont contribué à l'accomplissement de ce travail.

Dédicace

Je dédie tout d'abord ce travail à ma jolie grande famille , mes très chère parents pour leurs patience, encouragement et les valeurs de la vie qui m'ont enseigné pour aller toujours de l'avant, faire de mon mieux et surtout de ne jamais abandonner

A mes frères et sœurs qui ont contribué chacun à leurs manières durant ces années d'études à ma réussite

Egalement à ma deuxième famille les personnes qui m'entoure et qui m'enrichie chaque jours avec qui je partage mon quotidien difficile ou joyeux soit-il, merci de votre soutiens et de votre présence

Enfin je remercie Nesrine un binôme pas comme les autres, courageuse, généreuse et une force de caractère qui m'a beaucoup motivé et stimulé durant ces derniers mois

BABA AHMED Farah

Dédicace

Je dédie ce mémoire à ma très chère famille, tout d'abord ma mère pour son soutien, ses encouragements sans cesse, et son aide précieuse, mon père pour la zen attitude qu'il dégage, qui m'inspire, et qui me redonne confiance en moi, mon mari pour sa patience et sa compréhension, ses encouragements, à mes deux frères Amine et Karim qui ont toujours été là pour moi, pour leurs aides et soutiens.

A notre encadreur Mme HALFAOUI Amel, qui nous a guidés tout au long de ce travail, qui n'a pas toujours été de tout repos.

A mon amie Nawel qui n'a pas cessé de m'encourager, à une personne assez spéciale pour moi Nazima pour ce qu'elle est pour moi, à mes collègues à la banque BADR.

Enfin pour l'adorable personne qui a partagé ce travail avec moi qui est assez remarquable, quelqu'un d'acharné, d'une grande persévérance et grâce à laquelle, nous avons réussi à mener à bien ce travail.

Et pour n'oublier personne, j'utiliserai la formule :

« Merci à ... »

MAHMOUDI Nesrine

Table des matières

Introduction générale :	1
Chapitre I :	2
Concepts et notions de base d'un entrepôt de données	2
I.1 Introduction	3
I.2 Définition business intelligence (BI)	3
I.3 Entrepôts de données	4
I.3.1 Définition d'un entrepôt de données	4
I.3.2 Les caractéristiques	5
I.3.3 Modélisation multidimensionnelle d'un entrepôt de données	6
I.3.3.1 Le schéma en étoile	7
I.3.3.2 Schéma en flocon de neige	8
I.3.3.3 Schéma en constellation	9
I.3.3.4 Schéma en cube	10
I.3.4 Opérations sur le cube	11
I.3.5 Architecture d'un entrepôt de données	12
I.4 Définition OLAP	12
I.5 Implémentation des entrepôts des données	13
I.5.1 Systèmes ROLAP (Relational OLAP) :	13
I.5.2 Systèmes MOLAP (Multidimensionnel OLAP) :	13
I.5.3 Systèmes HOLAP (Hybride OLAP) :	13
I.5.4 Base DOLAP (Desktop OLAP):	14
I.5.5 Base OOLAP (Object OLAP):	14
I.6 Etude des différents serveurs des entrepôts de données	14
I.7 Comparaison entre les entrepôts de données et les SGBD transactionnel	14
I.8 Restitution des données	16
I.8.1 Langage de requêtes	16
I.8.2 Reporting	18
I.8.3 Tableaux de bord	19
I.9 Conclusion	19
Chapitre II :	20
Conception et réalisation de l'entrepôt bancaire	20
II.1 Introduction	21

II.2 Définition de la banque BADR.....	21
II.3 Modélisation et alimentation de l'entrepôt	22
II.3.1 Modélisation multidimensionnelle	22
II.3.2 Alimentation.....	23
II.4 Conception du cube de données XML	25
II.5 Conclusion	27
Chapitre III :	28
Mise en place technique de l'entrepôt bancaire	28
III.1 Introduction	29
III.2 Architecture technique de l'entrepôt bancaire	29
III.2.1 Pentaho Data Integration.....	29
III.2.2 MySQL Workbench – Modélisation des données/Conception de MySQL.....	30
III.2.3 Mondrian SchemaWorkbench	31
III.2.4 Mondrian.....	31
III.2.5 Pentaho	32
III.3 Présentation des étapes de conception de notre entrepôt de données :.....	32
III.3.1 Création et alimentation de la Base de Données:	32
III.3.2 Création du cube	39
III.3.3 Exploitation du cube	41
III.3.4 Teste du tableau de bords	45
III.4 Conclusion	46
 Conclusion générale et perspectives	 47
 Références Bibliographique	 48
Liste des figures.....	50
Liste des tableaux.....	51
Résumé.....	52

Introduction générale :

Le processus décisionnel est un projet qui se construit, il est né d'un besoin exprimé par les entreprises à cause du volume important des données à manipuler et à analyser car le système traditionnel ne suffit plus pour pérenniser l'activité de cette dernière, car celui-ci convient bien aux applications gérant l'activité quotidienne de l'entreprise, mais s'avère inadapté au Décisionnel. En entreposant les données, le processus décisionnel apporte la solution au problème de la croissance continue des données provenant de différentes sources et de différents formats en les homogénéisant et en les organisant dans des data mart où elles sont historiées, résumées et consolidées. Le volume de données des entrepôts est important et va de centaines de giga-octets à des téraoctets, voir même encore davantage de nos jours.

Notre projet de fin d'étude a été réalisé dans le cadre d'une institution financière « BANQUE DE L'AGRICULTURE ET DU DEVELOPPEMENT RURAL ».

La BADR Bank, la première banque à réseaux au niveau national constitué de plus de 300 agences et 39 directions régional, un nombre assez important de clients, vue le secteur qu'elle développe.

En effet, il est nécessaire d'apporter une solution pour l'analyse de ce gros volume de données.

Bill Inmon dans son ouvrage de référence « using the data Warehouse » définit l'entrepôt de la façon suivante « l'entrepôt de données est une collection de données orientés sujet, intégrées, non volatiles et historiées, organisées pour support d'un processus d'aide à la décision ».

Nous avons choisi le Groupe Régional de Tlemcen 013 et ses agences rattachées pour réaliser notre entrepôt de données, pour faire une analyse sur le nombre de crédits octroyés dans ces régions,

Notre mémoire est organisé selon le plan suivant :

Dans le premier chapitre, nous définissons des notions et des concepts importants sur les entrepôts de données, décrivant la méthode de modélisation et d'implémentation.

Dans le deuxième chapitre, nous présentons les étapes détaillées de la conception et la mise en place de notre entrepôt de données de l'extraction à la publication.

Dans le dernier chapitre, nous expliquons la mise en place technique en définissant les outils utilisés, en y joignant des captures d'écran de notre système.

Chapitre I :

Concepts et notions de base d'un entrepôt de données

I.1 Introduction

Aujourd'hui l'informatique ne cesse de se généraliser dans tous les secteurs d'activité, ce qui conduit les entreprises à conserver des sources d'information volumineuse et complexe, d'où la nécessité d'analyse et de consolidation de ces dernières pour une meilleure prise de décision.

Face à ce réel besoin, les entreprises ont recours à des systèmes d'aide à la décision spécifiques, basés sur l'approche des entrepôts de données qui ont fait leurs apparitions à la fin des années 1980.

Dans ce chapitre, nous allons présenter les concepts et notions de base des entrepôts de données.



Figure I-1 Modèle du processus de prise de décision [1]

I.2 Définition business intelligence (BI)

Business Intelligence «BI» ou L'informatique décisionnelle se définit par les différents outils, moyens et méthodologies qui permettent la collecte de différentes sources de

données brutes afin de les transformer en informations exploitable pour y appliquer des requêtes. On en tire alors diverses vues, ou modes de visualisation, tels que des rapports et des tableaux de bord, pour mettre les résultats analytiques à la disposition des décideurs et des acteurs de l'entreprise [2].

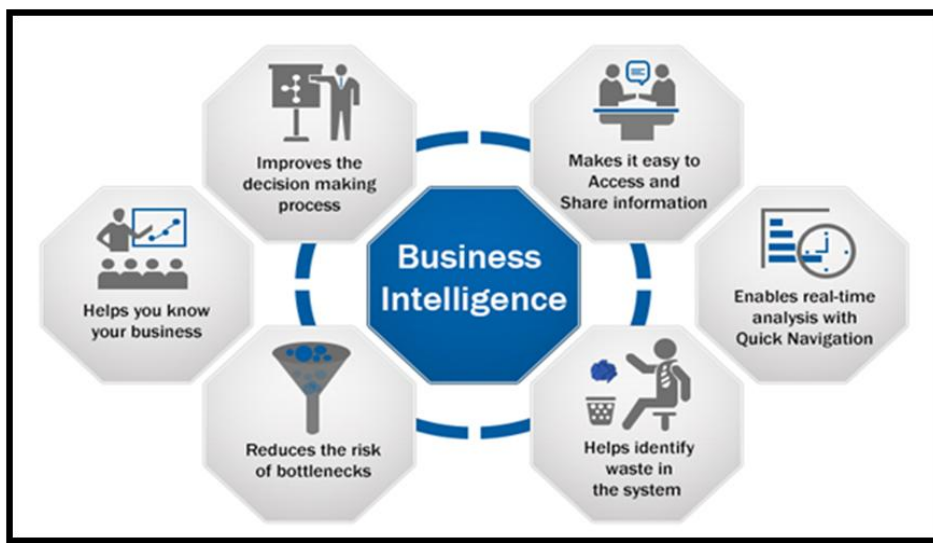


Figure I. 2 business intelligence [21]

Commentaire [A1]: Référencer l'image
sauf si c'est vous qui l'avez faite

I.3 Entrepôts de données

I.3.1 Définition d'un entrepôt de données

Père du concept, Bill Inmon dans son livre "Building the Data Warehouse" (John Wiley and Son 1996) le décrit ainsi : "Un entrepôt de données est une collection de données thématiques, intégrées, non volatiles, historiées et exclusivement destinées aux processus d'aide à la décision" par la suite nous allons détailler ces caractéristiques [4].

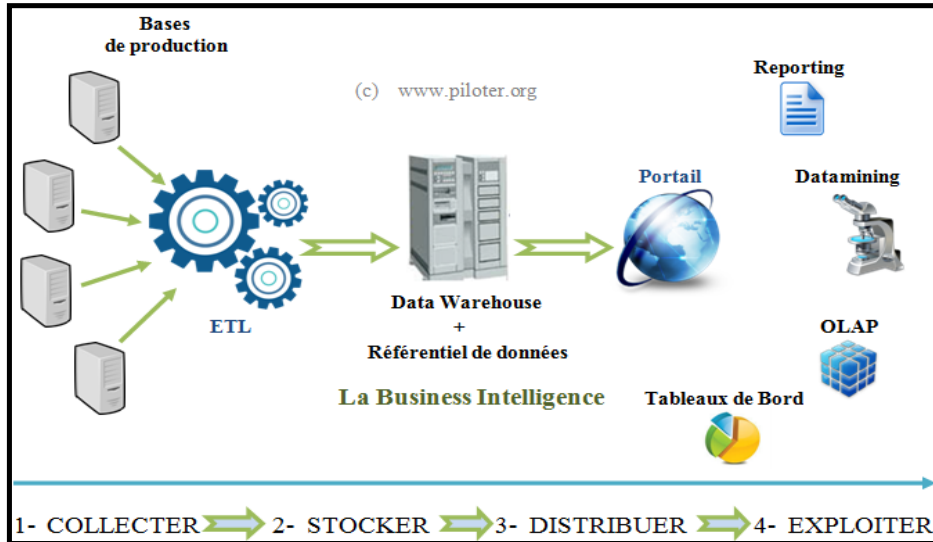


Figure I. 3 Entrepôt de données [5]

I.3.2 Les caractéristiques

L'Entrepôt de données n'est pas une simple copie des données de production mais il est également organisé et structuré.



Figure I. 4 Caractéristiques d'un data warehouse

- **Orienté sujet :**

Les données des entrepôts sont organisées par sujet plutôt que par application, les ventes par exemple, seront rapatriées des différentes bases OLTP de production et regroupées.

- **Intégrées :**
Les données provenant des différentes sources sont intégrées avant d'être proposées à l'utilisation. L'intégration (mise en correspondance des formats, par exemple), permet d'avoir une cohérence de l'information.
- **Non volatiles :**
A la différence des données opérationnelles, celles de l'entrepôt ne disparaissent pas et ne changent pas au fil des traitements, au fil du temps (Read-Only). Le rafraîchissement de l'entrepôt, consiste à ajouter de nouvelles données, sans modifier ou perdre celles qui existent.
- **Historiés :**
Les données non volatiles sont aussi horodatées. On peut ainsi visualiser l'évolution dans le temps d'une valeur donnée. Le degré de détail de l'archivage est bien entendu relatif à la nature des données. Toutes les données ne méritent pas d'être archivées.

I.3.3 Modélisation multidimensionnelle d'un entrepôt de données

La construction d'un entrepôt de données est réalisée à partir de la modélisation multidimensionnelle. [6]

Les données de la modélisation multidimensionnelle sont organisées de manière à mettre en évidence le sujet de l'analyse (la table des faits) et les différentes perspectives de l'analyse (les tables des dimensions) qui symbolisent les différentes valeurs de l'activité analysée.

Par exemple, dans le fait Ventes (Figure I.5), nous pouvons avoir la mesure "Quantité de produits vendus par magasin". Une dimension indique une perspective de l'analyse. Elle se constitue d'attributs qui enregistrent les descriptions textuelles. Les paramètres textuels peuvent être une source des restrictions dans une requête. Par exemple, dans la requête "Quantité de produits vendus dans la région Tlemcen durant le mois de septembre 2011". Nous trouvons les paramètres : région "Tlemcen " et mois "septembre 2011". Une hiérarchie représente les paramètres d'une dimension selon leur niveau de granularité ou de détail. Les paramètres sont ordonnés par une relation "est_plus_fin" et notée $P1 \rightarrow P2$. Dans notre exemple sur les ventes (cf. figure 1.1) nous pouvons avoir une hiérarchie pour la dimension Magasin de la forme suivante : Commune $\rightarrow$ Département $\rightarrow$ Région $\rightarrow$ Pays

I.3.3.1 Le schéma en étoile

Ce model tire son nom de sa configuration, en effet il se forme d'un objet central appelé table des faits et est relié à un ensemble d'objets de manière radiale, les « tables de dimension».

La table des faits :

- contient les différentes mesures
- Possède une clé étrangère pour chacune des tables de dimension.
- est dé-normalisées.

Les tables des dimensions :

- contiennent les attributs définissant chacun des membres des niveaux hiérarchiques d'une dimension.
- contiennent une clé primaire.
- sont dé-normalisées.

La figure I.5 représente le schéma en étoile en exposant les ventes effectuées au cours d'un jour dans les différents magasins de l'entreprise. Dans ce cas, au centre une table de faits appelée Ventes et autour les différentes tables de dimensions : Temps, Produit et Magasin.

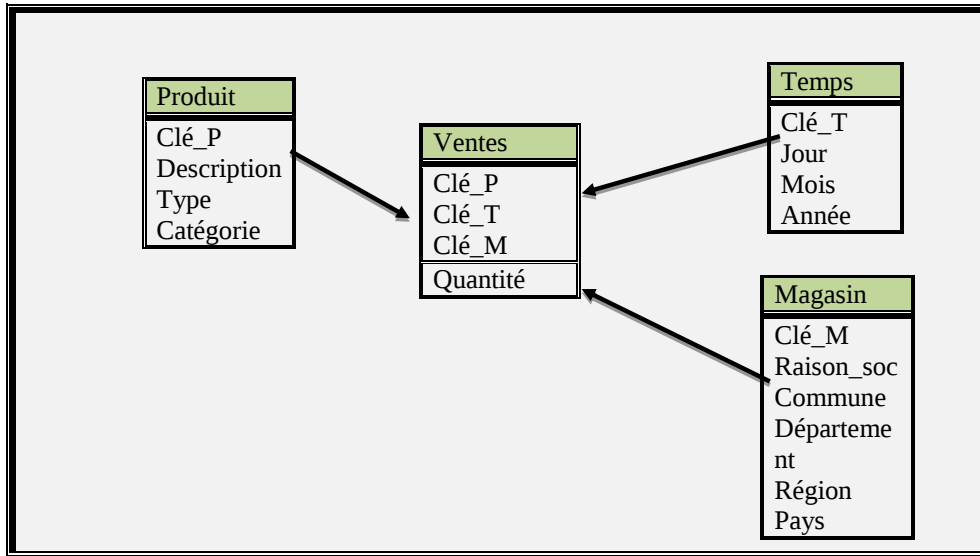


Figure I. 5 Schéma en étoile

Ce modèle est recommandé pour ces avantages qui sont sa faible complexité, sa facilité de compréhension pour l'utilisateur final et pour les liens directs avec les structures logiques des données.

I.3.3.2 Schéma en flocon de neige

Le schéma en flocon est dérivé du schéma en étoile, il se compose d'une relation centrale entourée des différentes tables de dimension qui sont normalisées. Avec ce schéma, les dimensions sont éclatées ou décomposées en sous hiérarchies. L'expression « hiérarchies alternatives » est utilisée pour désigner les hiérarchies différentes et parallèles d'une même dimension.

Les hiérarchies pour le schéma en flocon de neige de l'exemple de la figure I.5 sont :
Dimension Temps = Jour → Mois → Année

Dimension Magasin = Commune → Département → Région → Pays

La figure I.6 montre le schéma en flocon de neige avec les dimensions Temps et Magasin éclatées en sous hiérarchies (hiérarchie alternative), la dimension Temps a été éclatée en deux hiérarchies alternatives, Temps et T_Mois. La deuxième dimension Magasin, a été décomposée en trois hiérarchies alternatives : Magasin, M_Département et M_Région.

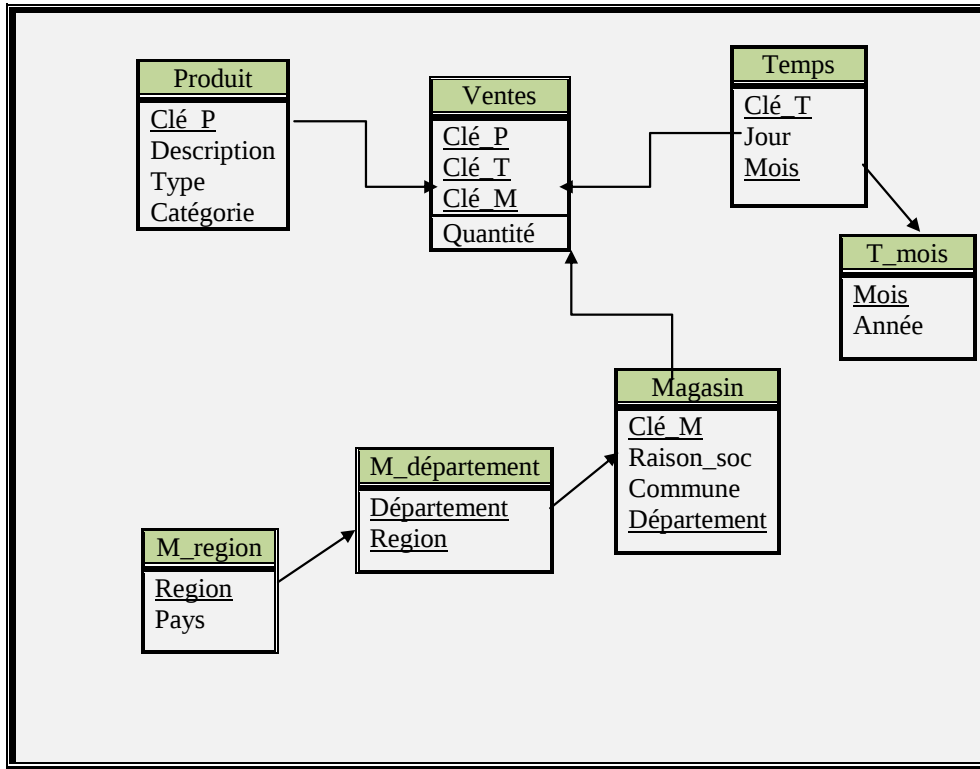


Figure I. 6 Schéma en flocon de neige

Ce modèle minimise le volume des tables à joindre lors de l'exécution des requêtes par rapport au schéma en étoile mais le modèle de données est plus complexe que le schéma en étoile car comportant un nombre plus élevé de tables.

1.3.3.3 Schéma en constellation

Le schéma en constellation dérive du schéma en étoile, il représente plusieurs relations de faits partageant ou non des dimensions.

La **figure I.7** montre le schéma en constellation qui est composé de deux relations de faits. La première s'appelle Ventes et enregistre les quantités de produits qui ont été vendus dans les différents magasins pendant un certain jour. La deuxième relation gère les différents produits achetés aux fournisseurs pendant un certain temps.

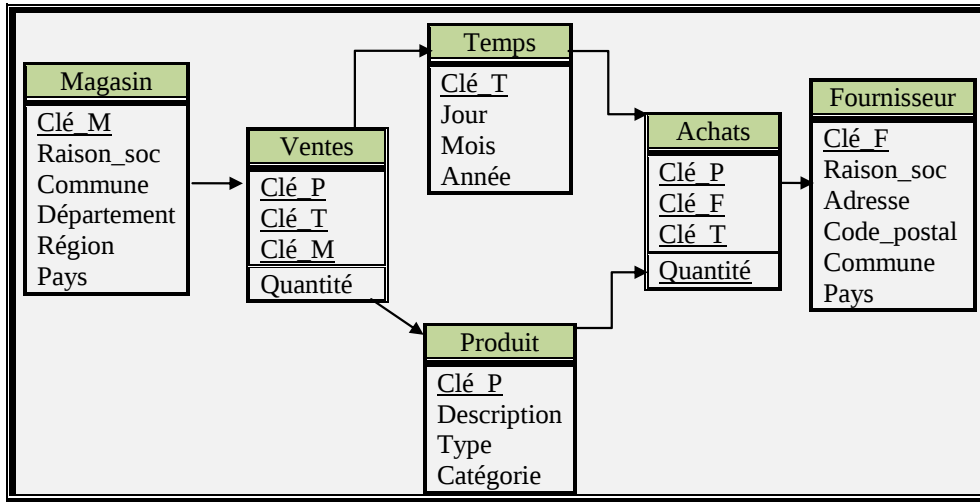


Figure I. 7 Schéma en constellation

La relation de faits **Ventes** partage les dimensions **Temps** et **Produits** avec la table **Achats**. Néanmoins, la dimension **Magasin** appartient seulement à **Ventes**. Egalement, la dimension **Fournisseur** est liée seulement à la relation **Achats**.

I.3.3.4 Schéma en cube

Dans le modèle multidimensionnel, le cube est l'objet central, lequel est composé des éléments nommés cellules qui contiennent une ou plusieurs mesures. La reconnaissance de la cellule est faite à travers les axes, ou chaque axe correspond à une dimension. La dimension est constituée de membres qui représentent les différentes valeurs.

En reprenant une partie du schéma en étoile de la figure I.5, nous pouvons construire le schéma multidimensionnel suivant.

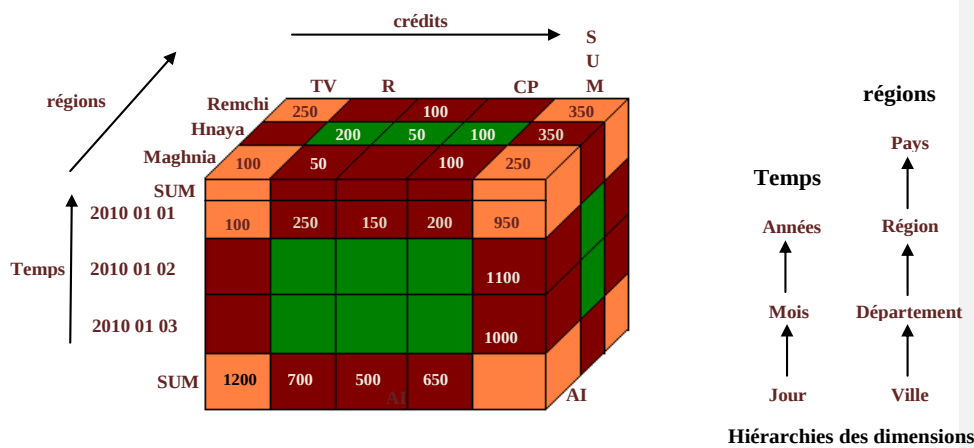


Figure I. 8 Schéma en cube

La figure I.8, présente un schéma multidimensionnel pour les ventes qui ont été réalisées dans les magasins pour les différents produits au cours d'un temps donné (jour). Par exemple, nous avons la quantité de 100 Téléviseurs vendus dans le magasin de Maghnia pendant le 1er janvier 2010.

I.3.4 Opérations sur le cube

- **Rotate:** sélection du pivot d'analyse en faisant tourner le cube;
- **Slice:** extrait une tranche du cube, c'est-à-dire, un sous-ensemble des valeurs du cube correspondant à une certaine valeur d'une des dimensions non-découpées; [12]
- **Drill-down:** détaille les données en descendant le niveau hiérarchique d'une dimension;
- **Roll-up:** agrège les données en augmentant le niveau hiérarchique d'une dimension;

I.3.5 Architecture d'un entrepôt de données

La phase essentielle d'un entrepôt de données est l'alimentation des données provenant des bases de production. Afin d'effectuer cette phase l'utilisation d'outils logiciels est nécessaires. On parle d'ETL (Extract, Transform, Load). [7]

L'alimentation d'un entrepôt est organisée selon les phases suivantes :

- **Découverte des données** : Identification des données à importer dans l'entrepôt de donnée qui se trouve dans les systèmes sources.
- **Extraction des données** : Collection des données importantes dans les systèmes de production.
- **Transformation des données** : Rendre les données cohérentes avec la structure d'un entrepôt de données c'est-à-dire les données des systèmes de production doivent être agrégées ou calculées avant leur chargement.
- **Chargement des données** : Insertion des données au sein de l'entrepôt de données.

I.4 Définition OLAP

OLAP (On Line Analytical Processing) est défini comme “ une catégorie de logiciels axées sur l'exploration et l'analyse rapide des données selon une approche multidimensionnelle a plusieurs niveaux d'agrégation.” [Caron1998]. L'architecture d'un système OLAP est constituée de trois services : la base de données, le serveur et le module client [Bédard et *al.* 1997]. [8]

- La base de données

- Les données qu'elle contient peuvent provenir d'un entrepôt de données.
- Elle doit pouvoir supporter des données agrégées.
- Elle doit posséder une structure multidimensionnelle.

- Le serveur

Gère la structure multidimensionnelle dans le SGDB et l'accès des utilisateurs à celle-ci, il est conçu spécifiquement pour manipuler des données structurées de façon multidimensionnelle.

- Le module client

Est un logiciel d'accès, qui permet à l'utilisateur de manipuler et d'explorer les données de nature multidimensionnelle. Il permet aussi de visualiser des données telles que les diagrammes à barres ou les diagrammes circulaires, pour ne nommer que les plus courants.

Les objectifs attendus par l'utilisation d'OLAP sont les suivants :

- Assistance pour une analyse optimale des données sans se focaliser sur les moyens utilisés.
- Rapidité et facilité.
- Visualisation multidimensionnelle des données.

I.5 Implémentation des entrepôts des données

Il existe différentes stratégies d'implémentation d'ED et d'analyse OLAP, Ces implémentations reprennent toutes les concepts OLAP. Parmi ces implémentations nous citons les plus utilisés : [10]

I.5.1 Systèmes ROLAP (Relational OLAP) :

- Utilisent un SGBD relationnel classique avec des adaptations spécifiques à l'OLAP.
- La base relationnelle de l'entrepôt est organisée pour réagir comme une base OLAP.
- Lents et peu performants mais sans limites de taille.

I.5.2 Systèmes MOLAP (Multidimensionnel OLAP) :

- Utilisent un SGBD multidimensionnel (MOLAP), ils sont l'application physique du concept OLAP (réellement d'une structure multidimensionnelle).
- Très rapides et performants mais limité au gigaoctet.

I.5.3 Systèmes HOLAP (Hybride OLAP) :

- C'est un compromis :
 - Une base MOLAP pour les données souvent consultées (la minorité),

- Une base ROLAP pour les autres (la majorité).

I.5.4 Base DOLAP (Desktop OLAP):

Base OLAP très limitée en taille, hébergée sur le poste client, et très rapide.

I.5.5 Base OOLAP (Object OLAP):

Utilise un SGBD Orienté Object : peu utilisé.

I.6 Etude des différents serveurs des entrepôts de données

Nous allons présenter un ensemble de serveurs existant et les classer par appartenance à l'implémentation qui a été présentée plus haut [10]

- IBM DB2 UDB Server: ROLAP
- Jedoxolap server: MOLAP HOLAP ROLAP
- Oracle Express-server: MOLAP ROLAP
- Mondrian OLAP server :ROLAP
- Microsoft SQL Server : HOLAP

I.7 Comparaison entre les entrepôts de données et les SGBD transactionnel

Dans la réalisation d'une structure informatique décisionnelle on pourrait s'interroger pourquoi ne pas utiliser un simple SGBD. [11]

La raison est que les objectifs, la manière de stocker les données ainsi que les requêtes d'un SGBD sont différentes de celles d'un entrepôt de donnée. Ils sont ainsi basés sur deux systèmes différents : OLTP et OLAP

OLTP

Le modèle qui utilise les SGBD est OLTP (On Line Transaction Processing). Le mode de traitement est transactionnel. L'objectif d'utilisation d'un tel système est d'insérer, modifier et interroger rapidement la base de données en toute sécurité. Ces actions

doivent pouvoir être effectuées en temps réel par de nombreux utilisateurs en simultané.

Chaque opération travail sur de petites quantités d'informations, sur les versions les plus récentes des données.

OLAP

Les entrepôts de données utilisent le système OLAP (On Line Analytical Processing). Le travail de ce type de système se fait uniquement en lecture. Pour effectuer l'analyse, les programmes consultent une grande quantité de données. Les principaux objectifs sont de regrouper et d'organiser des informations à partir de différentes sources, les intégrer et les stocker afin de donner à l'utilisateur une vue axée métier, récupérer et analyser l'information facilement et rapidement. Ceci exige la consultation des versions historiques de la base ce qui permet d'ignorer dans une durée limitée les nouvelles mises à jour. Ces bases sont généralement d'un ordre de grandeur considérablement plus haut que celui des bases de données OLTP, en raison de la rétention d'informations historiques.

Comparaison

Caractéristiques	OLTP	OLAP
Utilisation	SGBD (base de production)	Datawarehouse
Opération typique	Mise à jour	Analyse
Type d'accès	Lecture écriture	Lecture
Niveau d'analyse	Elémentaire	Global
Quantité d'informations échangées	Faible	Importante
Orientation	Ligne	Multidimensionnel
Taille BD	Faible (max quelques GB)	Importante (pouvant aller à plusieurs TB).
Requêtes	Simple requêtes retournant quelques enregistrements (lignes) de la BD	Requêtes complexes impliquant souvent plusieurs tables et faisant l'agrégation de valeurs
Ancienneté des données	Récente	Historique
Temps de réponse	Instantané	Quelques seconds à minute

Tableau I. 1 Comparaison entre OLTP et OLAP

Le tableau ci-dessus récapitule des différences entre OLTP et OLAP.

Les bases OLTP répondent à des questions simples rapidement, exemple : les ventes du produit X. Les bases OLAP répondent des requêtes plus complexes : les ventes du produit X par vendeur, région et par mois.

I.8 Restitution des données

La mise en œuvre du Data Warehouse est la dernière étape d'un projet Data Warehouse, soit son exploitation.

Il existe divers façons de naviguer au sein d'un cube OLAP afin de récupérer une information, pour se faire, un certain nombre d'outils graphiques sont disponibles, souvent spécifiques aux éditeurs du cube, qui offrent la possibilité de naviguer facilement au sein de celui-ci. [14].

I.8.1 Langage de requêtes

Comme les langages de requêtes SQL pour les bases de données, OLAP utilise aussi des langages de requêtes de calcul avec une syntaxe semblables aux tableurs. L'inconvénient est qu'il n'y a pas de langage universel. [8]

Néanmoins, on peut distinguer deux tendances :

- MDX de l'anglais Multidimensional Expressions intégré à Microsoft SQL Server 2005.
- OLAP DML (Data Manipulation Language) intégré à Oracle 10g.

Comparaison entre SQL et MDX

- MDX est fait pour naviguer dans les bases multidimensionnelles et pour exécuter des requêtes sur tous leurs objets (dimensions, hiérarchies, niveaux, membres et cellules) afin d'obtenir (simplement) une représentation sous forme de tableaux croisés. [9]
- La syntaxe de MDX ressemble à celle de SQL par ses mots clé SELECT, FROM, WHERE,

Mais leurs sémantiques sont différentes :

- SQL construit des vues relationnelles

-MDX construits des vues multidimensionnelles des données

- Analogies entre termes multidimensionnels (MDX) et relationnels (SQL) :

Relationnel (SQL)	Multidimensionnel (MDX)
Table	Cube
Colonne (chaîne de caractère ou valeur numérique)	Niveau (Level)
plusieurs colonnes liées ou une table de dimension	Dimension
Colonne (discrète ou numérique)	Mesure (Measure)
Valeur dans une colonne et une ligne particulière de la table	Membre de dimension (Dimension member)

Tableau I. 2 Comparaison entre SQL et MDX

Structure générale d'une requête :

- **SQL** : SELECT column1, column2, ..., columnn FROM table
- **MDX** : SELECT axis1 ON COLUMNS, axis2 ON ROWS FROM cube •
Clause

FROM spécifie la source de données

En SQL : une ou plusieurs tables

En MDX : un cube

SELECT indique les résultats que l'on souhaite récupérer par la requête :

En SQL :

- Une vue des données en 2 dimensions.
- Lignes (rows) et colonnes (columns), les lignes ont la même structure définie par les colonnes

En MDX :

- Nombre quelconque de dimensions pour former les résultats de la requête
- Terme d'axe pour éviter confusion avec les dimensions du cube

- Pas de signification particulière pour les rows et les columns, mais il faut définir chaque axe : axe1 définit l'axe horizontal et axe2 définit l'axe vertical

Le contenu des axes :

- Un membre (Member) = est un item dans une dimension et correspond à Un élément spécifique de donnée :

[Time]. [2012]

[Customers].[All Customers].[Mexico].[Mexico]

[Product].[All Products].[Drink]

- Un tuple= une collection de membres de différentes dimensions :

(([Time].[2012], [Product].[All Products].[Drink])

(2012, Drink)

(2012, [Customers].[All Customers].[Mexico].[Mexico])

- Un set = une collection de tuples :

I.8.2 Reporting

Destiné essentiellement à la production de rapports et de tableaux de bord, « il est la présentation périodique de rapports sur les activités et résultats d'une organisation, d'une unité de travail ou du responsable d'une fonction, destinée à en informer ceux chargés de les superviser en interne ou en externe, ou tout simplement concernés par ces activités ou résultats ». [15]

Ces outils de Reporting ne sont pas, à proprement parler, des instruments d'aide à la décision, mais, lorsqu'ils sont utilisés de manière appropriée, ils peuvent fournir une précieuse vue d'ensemble.

Les rapports sont alors créés par le biais d'outils de Reporting qui permettent de leur donner un format prédéterminé. Les requêtes sont constituées lors de l'élaboration des rapports qui seront ensuite diffusés périodiquement en automatique ou ponctuellement à la demande.

I.8.3 Tableaux de bord



Figure I. 9 Tableau de bord [16]

Un tableau de bord (ou « dashboard ») est un outil de pilotage clef dans une suite décisionnelle en effet il donne des informations synthétiques de l'évolution d'un processus facilement accessibles. « Le tableau de bord est un ensemble d'indicateurs peu nombreux conçus pour permettre aux gestionnaires de prendre connaissance de l'état et de l'évolution des systèmes qu'ils pilotent et d'identifier les tendances qui les influenceront sur un horizon cohérent avec la nature de leurs fonctions » [Bouquin, 2003].[15]

Cette forme de restitution résume l'essentiel, en mettant en évidence l'état d'un indicateur par rapport à un objectif, tout en adoptant une représentation graphique de l'information.

I.9 Conclusion

Dans ce chapitre, des notions et des concepts importants sur les entrepôts de données ont été présentés, décrivant la méthode de modélisation d'un entrepôt de donnée et les différentes stratégies d'implémentation ainsi que les différents modèles multidimensionnels pour sa construction, montrant ainsi l'impact de ces derniers sur la prise de décision au sein des entreprises.

Dans le prochain chapitre, nous traiterons les étapes de conception et de réalisation d'un entrepôt bancaire.

Chapitre II :

Conception et réalisation de l'entrepôt bancaire

II.1 Introduction

Nous avons réalisé un entrepôt de données afin de permettre une aide à la décision au sein de la banque BADR se basant sur la région de Tlemcen et ses agences rattachées, vu le manque d'informations accessibles, notre analyse s'est basée principalement sur le processus d'affaire : Les Crédits. Le fait principale à analyser repose sur une mesure : le calcul du nombre de crédit octroyés en se basant sur des axes d'analyses régions, crédit, temps et client.

II.2 Définition de la banque BADR

La Banque de l'Agriculture et du Développement Rural est une institution financière nationale créée le 13 mars 1982, sous la forme juridique de société par actions. Son capital social est de 33000000000DA. Elle a pour missions principales le développement du secteur agricole et la promotion du monde rural.



Figure II. 1 Logo de la banque BADR

Constituée initialement de 140 agences, son réseau compte actuellement plus de 300 agences et 39 directions régionales. Quelques 7 000 cadres et employés activent au sein des structures centrales, régionales et locales. La densité de son réseau et l'importance de son effectif font de la BADR la première banque à réseau au niveau national.

II.3 Modélisation et alimentation de l'entrepôt

II.3.1 Modélisation multidimensionnelle

L'objectif principal des modèles multidimensionnels est de permettre un accès très performant aux données. Ainsi, les données sont disposées autour des faits que l'on souhaite analyser, caractérisés par des indicateurs appelés mesures, ils sont généralement de type numériques et additives, permettant la mesure de l'activité modélisée. Le modèle multidimensionnel est représenté par un schéma en étoile où les faits sont représentés par une table des faits reliée à un ensemble de tables de Dimension. [13]

Les avantages de ce modèle sont la facilité de navigation, grâce à la table de faits centrale, de bonnes performances en raison du faible nombre de jointures à effectuer pour l'analyse sur une dimension donnée et des agrégations faciles des mesures.

Citons le fait concernant les ventes des crédits (octroicredit) au sein de la banque (**figure II.2**), dont la mesure est le **nombre de crédits** et Les dimensions (clients, crédits, régions, temps) sont des descripteurs des faits de « octroicredit ». Ainsi, pour un client donné ou une région donnée, un crédit, une date, nous disposons du nombre de crédits octroyé à l'égard d'un client ou d'une région.

La figure ci-dessous montre le modèle multidimensionnel en étoile des données de l'entrepôt bancaire :

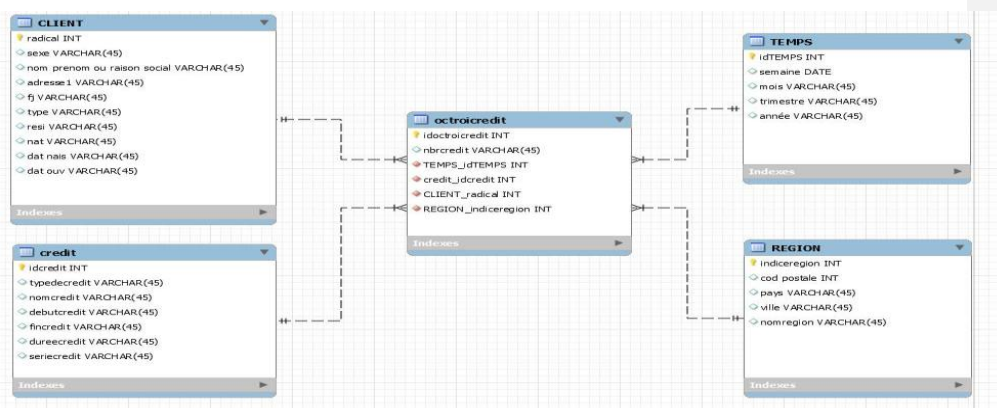


Figure II. 2 Modèle en étoile

II.3.2 Alimentation

L'alimentation d'un entrepôt de données se déroule essentiellement par la migration des données de la base de production vers l'entrepôt de données qui se fait à l'aide des outils ETL (Extracting, Transforming and Loading). Cette phase est très importante et constitue environ 80% du processus décisionnel, où les données sont sélectionnées, nettoyées transformées, ainsi l'entreposage de ces dernières facilitera la prise de décision. Pour notre projet, nous avons utilisé principalement un composant de l'outil d'un ETL permettant d'alimenter des tables historiées.

. Ce genre de composant est beaucoup utilisé dans les bases de données, suivant une modélisation dimensionnelle, pour alimenter des dimensions à variation lente,

Une transformation de dimension à variation lente coordonne la mise à jour et l'insertion d'enregistrements dans des tables de dimension d'entrepôts de données.

Prenons un exemple pour illustrer le paramétrage du composant, ce qui vaut toutes les explications.

Partons d'un fichier CSV simple comportant un client, ainsi que son adresse.

Nom prenoms ou raison scocial,Adresse,Date de naissance	
BENSABEUR DJILALI ,HAI EL NEDJMA N 107 IMMAMA ,5/22/1943	
BENSABEUR DJILALI,CITE LES OLIVIERS KIFFANE N 03,5/22/1943	

Figure II. 3 Exemple fichier CSV Client

On remarque que le client a changé d'adresse une fois. On voudrait qu'une fois en base de données, les informations soient historiées avec une date de début et une date de fin, pour avoir un intervalle de validité de l'adresse du client.

Nous avons donc une dimension client dans notre base de données représentée par la table « client »

Le but étant d'arriver au résultat suivant, pour la transformation du fichier CSV en une table dans une base de données :

Column Name	Datatype
dim-client	INT(11)
version	INT(11)
date to	DATETIME
date from	DATETIME
Nom prenom ou adresse social	VARCHAR(55)
Adresse	VARCHAR(55)

Figure II. 4 Exemple de table dans la base de données

Présentation de quelques tables de la base de production de la banque BADR que nous avons utilisé :

- Client (Radical ,Sexe, Nom Prénom ou Raison social , Adresse1 ,Adresse2 , Adresse3 ,FJ, Type, Resi, Nat,Date de naissance, Date d'ouverture)
- Crédit (Code, Typedecredit, Serie, Variation, Taux, Taux bonifié)
- Regions (indiceregion, code postal, pays, ville, nomregion)

Une fois les données traitées avec l'ETL les tables de notre entrepôt de données sont transformé comme suit :

- Client (Clet_Radical,version, dateto, datefrom, Radical, Sexe, Nom prénom ou raisons social, Adresse1, FJ, Type, Resi, Nat, Date de naissance, Date d'ouverture)
- Credit(clet_credit,version,dateto,datefrom,idcredit, type de crédit, nomcredit, seriecredit, taux ,taux bonifié)
- Regions (clet_region,version,dateto,date from, indiceregion, code postal,pays,ville, nomregion)
- Temps(clet_temps, version, date to, date from, date, id, y,m,d,yw,w,q,wd,m_name,wd_name)
- Octroiecredit (Clet_Radical, clet_credit, clet_region ,clet_temps, Nbr de credit,Radical,idcredit, indiceregion, id)

La table temps a été implémenté grâce à l'exécution d'un script « dimension temps »

II.4 Conception du cube de données XML

Un des principaux points de la Business Intelligence est l'analyse multidimensionnelle. Il devient primordial de pouvoir étudier une même information suivant différents axes. Par exemple, il est très intéressant pour la banque Badr de connaître le nombre de crédit octroyé par période temporelle, par régions, par client...c'est pourquoi le moyen pour effectuer ces analyses est d'utiliser des cubes (ou hyper cube).

L'illustration ci-dessous montre la conception du cube effectué par la rédaction d'un fichier XML, qui permettra de lier la table de fait, les dimensions, les hiérarchies et la mesure à la base de donnée

```

▪ <Cube name="cubeprojet" visible="true" cache="true" enabled="true">
▪ <Table name="octroicrodit">
▪ </Table>
▪ <Dimension type="StandardDimension" visible="true" foreignKey="clet_radical"
highCardinality="false" name="client">
▪ <Hierarchy name="client" visible="true" hasAll="true" primaryKey="clet_radical">
▪ <Table name="client">
▪ </Table>
▪ <Level name="nom prenomou raison social" visible="true" table="client" column="Nom
prenomou raison social" type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
▪ <Property name="adresse" column="Adresse1" type="String">
▪ </Property>
▪ <Property name="fj" column="Fj" type="String">
▪ </Property>
▪ <Property name="type" column="Type" type="Integer">
▪ </Property>
▪ <Property name="resi" column="Resi" type="Integer">
▪ </Property>
▪ <Property name="nat" column="Nat" type="Integer">
▪ </Property>
▪ <Property name="date nais" column="Datnais" type="String">
▪ </Property>
▪ <Property name="date ouv" column="Datouv" type="String">
▪ </Property>
▪ <Property name="version" column="version" type="Integer">
▪ </Property>
▪ <Property name="radical" column="Radical" type="String">
▪ </Property>
▪ <Property name="clet_radical" column="clet_radical" type="Integer">
▪ </Property>
▪ </Level>
▪ </Hierarchy>
▪ </Dimension>
▪ <Dimension type="TimeDimension" visible="true" foreignKey="clet_temps"
highCardinality="false" name="temps">
▪ <Hierarchy name="temps" visible="true" hasAll="true" primaryKey="clet_temps">
▪ <Table name="temps">
▪ </Table>
▪ <Level name="jours" visible="true" table="temps" column="date" type="String"
uniqueMembers="false" levelType="TimeDays" hideMemberIf="Never">

```

```

▪ <Property name="id" column="id" type="Integer">
▪ </Property>
▪ <Property name="version" column="version" type="Integer">
▪ </Property>
▪ </Level>
▪ </Hierarchy>
▪ </Dimension>
▪ <Dimension type="StandardDimension" visible="true" foreignKey="clet_credit"
highCardinality="false" name="credit">
▪ <Hierarchy name="credit" visible="true" hasAll="true" primaryKey="clet_credit">
▪ <Table name="credit">
▪ </Table>
▪ <Level name="nom credit" visible="true" table="credit" column="Nomcredit" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
▪ <Property name="seriecredit" column="Seriecredit" type="Integer">
▪ </Property>
▪ <Property name="type de credit" column="Typedecredit" type="String">
▪ </Property>
▪ <Property name="idcredit" column="idCredit" type="Integer">
▪ </Property>
▪ <Property name="taux" column="idCredit" type="Numeric">
▪ </Property>
▪ <Property name="tauxbonifi&#233;" column="Tauxbonifie" type="Numeric">
▪ </Property>
▪ </Level>
▪ </Hierarchy>
▪ </Dimension>
▪ <Dimension type="StandardDimension" visible="true" foreignKey="clet_region"
highCardinality="false" name="regions">
▪ <Hierarchy name="regions" visible="true" hasAll="true" primaryKey="clet_region">
▪ <Table name="regions">
▪ </Table>
▪ <Level name="nom region" visible="true" table="regions" column="Nomregion" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
▪ <Property name="code postale" column="Codepostale" type="Integer">
▪ </Property>
▪ <Property name="pays" column="pays" type="String">
▪ </Property>
▪ <Property name="ville" column="ville" type="String">
▪ </Property>
▪ <Property name="indiceregion" column="Indiceregion" type="Integer">
▪ </Property>
▪ <Property name="version" column="version" type="Integer">
▪ </Property>
▪ </Level>
▪ </Hierarchy>
▪ </Dimension>
▪ <Measure name="nombre de credit" column="NBRCREDIT" datatype="Integer"
aggregator="sum" visible="true">
▪ </Measure>
▪ </Cube>

```

II.5 Conclusion

Ce chapitre étale essentiellement les étapes de modélisation de l'entrepôt qui se fait dans tous les cas grâce à la modélisation dimensionnelle et l'alimentation des données qui constitue l'étape la plus fastidieuse et consommatrice en temps de notre projet.

Une fois l'alimentation terminée, l'exploitation des données peut alors se faire par différentes méthodes. L'utilisation d'outil OLAP reste, cependant, l'aspect le plus intéressant dans cette exploitation permettant la navigation dans les données de l'entrepôt à la demande.

Chapitre III :

Mise en place technique de l'entrepôt bancaire

III.1 Introduction

Ce dernier chapitre est consacré à la présentation des différents outils qui ont permis la réalisation de notre application et l'ensemble des fonctionnalités qu'offre cette dernière sous forme de prises d'écran avec des descriptions pour bien comprendre chaque étape de la réalisation et de la mise en place de notre entrepôt de donnée

III.2 Architecture technique de l'entrepôt bancaire

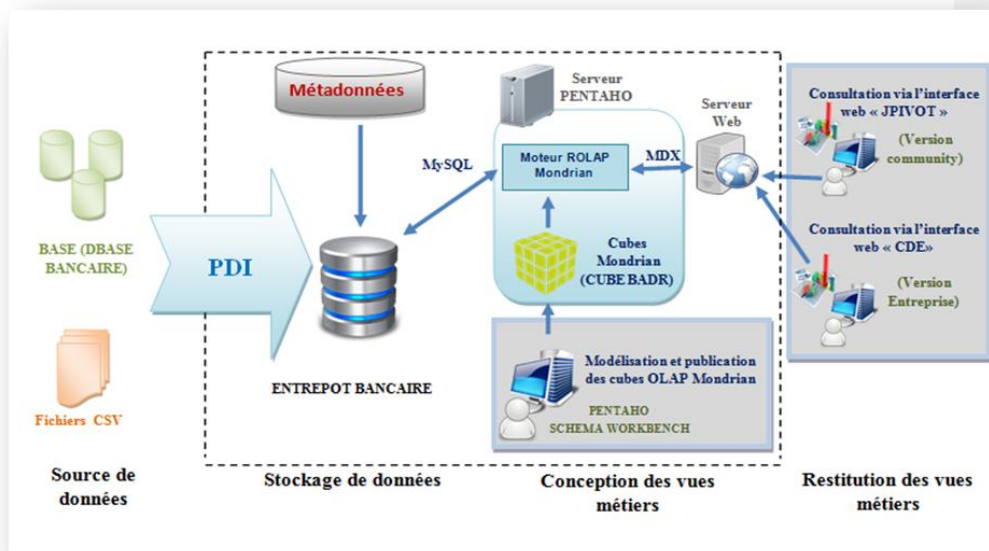


Figure III. 1 Architecture technique de l'entrepôt bancaire

III.2.1 Pentaho Data Integration

Pentaho Data Integration (PDI) permet aux entreprises d'extraire les données, à partir des sources complexes et hétérogènes et de créer des informations cohérentes et de bonne qualité pour les applications critiques.

Grace à une interface utilisateur graphique complète et un moteur de traitement parallèle, PDI offre une solution ETL (extraction, transformation, chargement) de haute

performance qui couvre tous les besoins en intégration de données. Les cas d'utilisation de PDI comprennent

- La Business Intelligence et l'entreposage de données
- La migration des données
- La consolidation des applications
- La synchronisation des données

PDI se compose de 3 applications :

- SPOON est l'outil qui permet grâce à son interface graphique de créer des transformations, les exécuter et les sauvegarder. Les composants permettant la manipulation des données sont nommés « étapes » (steps en anglais). Par exemple il existe une étape permettant d'extraire des données de diverses bases de données, une autre aidant à l'extraction depuis des fichiers. SPOON comprend un grand nombre d'étapes.
- Pan : application en ligne de commande permettant de lancer l'exécution d'une transformation donnée.
- Kitchen : application en ligne de commande permettant de lancer l'exécution d'une tâche donnée.

Dans notre cas nous avons utilisés Spoon

III.2.2 MySQL Workbench – Modélisation des données/Conception de MySQL

MySQL Workbench est un outil de conception et de modélisation visuelle des bases de données, proposé par MySQL/Sun. Il permet aux développeurs et aux DBA de créer de nouveaux modèles de données physiques pour les bases de données MySQL, mais aussi de modifier les bases de données physiques MySQL existantes avec une rétro/pro-ingénierie et des fonctions de gestion des changements. [17]

MySQL Workbench a été conçu pour optimiser la productivité et la réussite de l'utilisateur dans la conceptualisation, la communication, la création et la gestion des principales métadonnées d'entreprise, ainsi que des bases de données et des entrepôts de données à haute performance. L'interface et les processus automatisés de MySQL

Workbench garantissent une réussite immédiate pour les différents publics composant les entreprises modernes, notamment les administrateurs de bases de données, les développeurs d'applications, les architectes de données et les responsables informatiques. Le produit s'exécute sous Windows, Linux et Mac pour que les utilisateurs puissent concevoir leurs bases de données à partir de tous les systèmes d'exploitation majeurs. Un bref aperçu de toutes les fonctions de MySQL Workbench permet de présenter la façon dont l'outil offre les avantages décrits précédemment dans le domaine de la gestion des données basée sur des modèles.

II.2.3 Mondrian SchemaWorkbench

Le Mondrian SchemaWorkbench est une interface de conception qui vous permet de créer et de tester Mondrian OLAP schémas de cube visuellement. Le moteur Mondrian traite MDX requêtes avec le ROLAP (Relational OLAP) schémas. Ces fichiers de schéma sont des modèles de métadonnées XML qui sont créés dans une structure spécifique utilisé par le moteur Mondrian. Ces modèles XML peuvent être considérés comme des structures cubiques qui utilisent FACT et DIMENSION tableaux trouvés dans votre SGBDR existantes. Il ne nécessite pas qu'un cube physique est construit ou maintenu; seulement que le modèle de métadonnées est créé.[18]

III.2.4 Mondrian

Mondrian est un moteur OLAP (Online Analytical Processing) écrit en Java par Julian Hyde qui permet la conception, la publication et le requêtage de cubes multidimensionnels.

Mondrian permet l'exécution de requêtes en langage MDX sur des entrepôts de données s'appuyant sur des SGBDR, d'où sa caractérisation de « ROLAP » (Relational OLAP). En matière de ROLAP, Mondrian est la référence open source.

Mondrian permet d'accéder aux résultats dans un format multidimensionnel compréhensible par une API de présentation côté client, le plus souvent en mode Web, avec par exemple JPivot, Pentaho Analyzer, PentahoAnalysisTool, GeoAnalysisTool (G.A.T.)

Mondrian s'appuie sur une modélisation OLAP standard et peut donc se connecter à n'importe quel entrepôt de données conçu dans les règles de l'art de la Business Intelligence.

Il est intéressant de noter que Mondrian est le composant OLAP utilisé par la plupart des suites de BI Open Source notamment Pentaho, JasperServer et SpagoBI.[19]

III.2.5 Pentaho

Pentaho est une plate-forme décisionnelle open source complète possédant les caractéristiques suivantes :[20]

- une couverture globale des fonctionnalités de la Business Intelligence :
 - ETL (intégration de données),
 - reporting,
 - tableaux de bord ("*Dashboards*"),
 - analyse *ad hoc* (requêtes à la demande),
 - analyse multidimensionnelle (OLAP) ;
- Pentaho est une suite décisionnelle open source commerciale qui reste très « ouverte ». Les différences fonctionnelles entre la version libre (***community edition***) et la version payante (***enterprise edition***) restent limitées. La version libre de Pentaho permet d'installer une plate-forme décisionnelle complète !

III.3 Présentation des étapes de conception de notre entrepôt de données :

La réalisation de notre entrepôt de données s'effectue en 4 étapes essentielles comme suit:

III.3.1 Création et alimentation de la Base de Données

Nous procédons à la création et à l'alimentation des tables pour la base de donnée préalablement créée intitulée « Projet » avec MySQLWorkbench via ETL (PentahoData

Intégration (spoon)), qui est L'étape la plus critique et la plus importantes pour l'alimentation de notre entrepôt.

L'extraction des données, leurs transformations ainsi que leurs chargements sont réalisés de la manière suivante :

Lors du lancement de ETL, on a le panneau « navigateur » qui nous permet de créer une transformation dans laquelle une « palette de création » apparaît qui donne la possibilité de faire une extraction de n'importe qu'elle type de fichier, dans notre cas c'est un fichier CSV.

Pour cela, nous allons créer une transformation avec Pentaho Data Integration (PDI). Cette transformation est très simple et ne contient que deux étapes. La première pour l'extraction des données du fichier CSV et la seconde pour l'insertion en base de données grâce au composant de chargement de dimension à variation lente. Ce qui donne ceci :

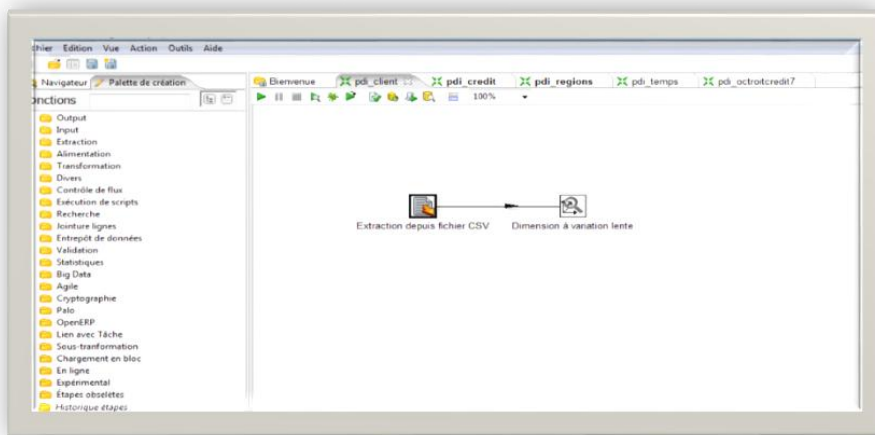


Figure III. 2 Extraction, transformation et chargement de la dimension client

La figure ci-dessus montre l'extraction et la transformation de la dimension Client cette étape se répétera pour les autres dimensions qui sont Crédit, Régions et Temps.

La figure ci-dessous montre la configuration effectuée pour l'extraction du fichier client

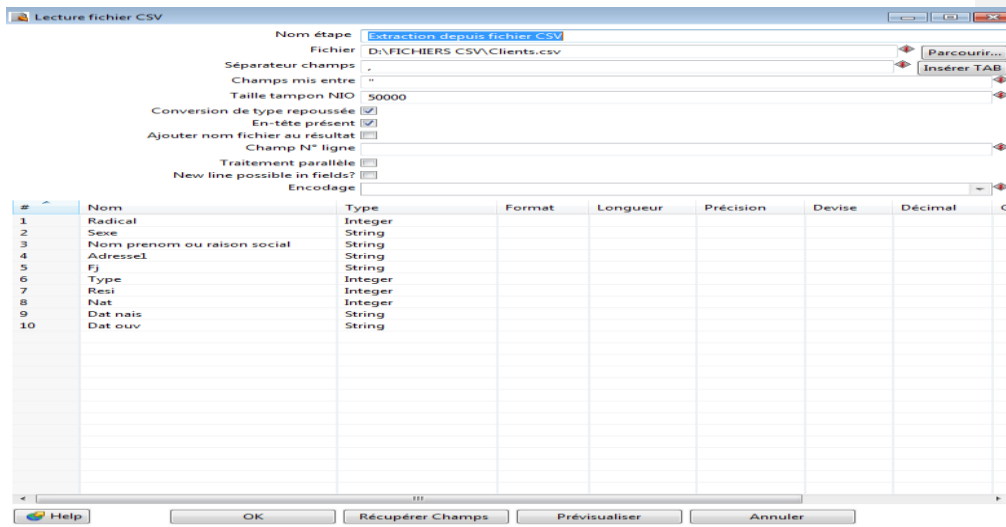


Figure III. 3 Extraction des données à partir d'un fichier CSV

Les 3 figures suivantes (figure3), (figure4) et (figure5) montrent la configuration qui se réalise en trois étapes distinctes.

La première concerne la configuration générale du composant, connexion à la base de données, table de la base à remplir, plus divers champs de configuration pour la taille du commit en base, etc.

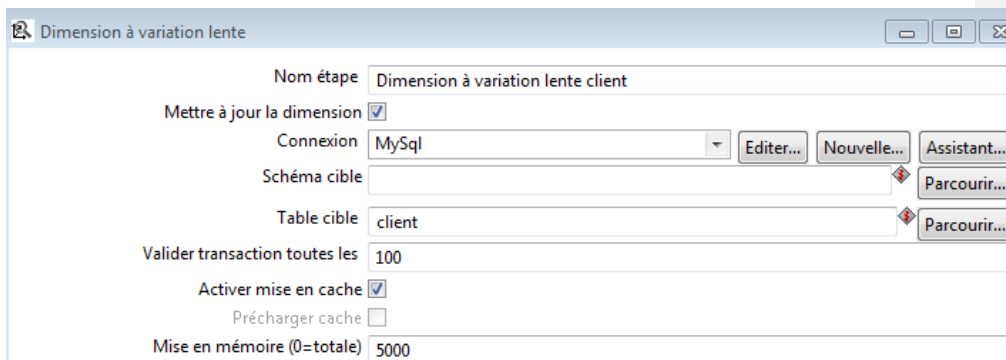


Figure III. 4 Configuration générale

La deuxième partie se décompose en deux onglets. Le premier permet la configuration du ou des champs clé. Dans notre cas le champ clé est « radical » en base de données

Figure III. 5 Configuration du champ clé

L'onglet numéro deux lui permet de configurer les champs que le composant doit surveiller pour détecter un changement à historique.

Figure III. 6 Configuration champs sujet aux changements

La dernière partie de la configuration permet la gestion de la clé technique de la base de données pour l'incrémentation de celle-ci. On peut gérer l'incrémentation de plusieurs façons, le maximum incrémenté, l'utilisation d'une séquence en base, ou un champ auto-incrémenté. Dans notre exemple, la clé technique est « celt_radical » et nous allons utiliser la gestion par maximum incrémenté.

Cette partie permet de choisir le champ représentant la version. Pour notre cas, ce sera le champ « version ».

Et enfin, nous avons une série de champs à configurer pour la gestion des dates. La première étape est de choisir le champ du flux représentant la date d'effet, ainsi que le champ de la date de début et date de fin. Il nous faut aussi définir l'année de la date minimale et la date maximale, dans notre cas, nous prenons l'an 1900 pour la date minimale et 2199 pour la date maximale.

Il existe aussi plusieurs autres façons de gérer la date de début, comme prendre la date courante du système, la date de démarrage du script, la valeur d'un champ date de la table, etc...

Les deux figures suivantes (figure 6) et (figure7) expliquent respectivement la création et l'alimentation de la table des faits octroi crédit à partir des dimensions

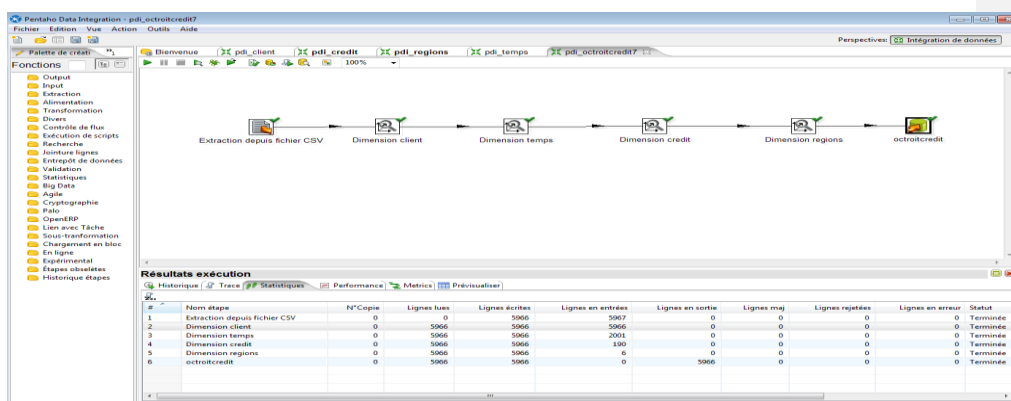


Figure III. 7 Création de la table des faits octroi crédit

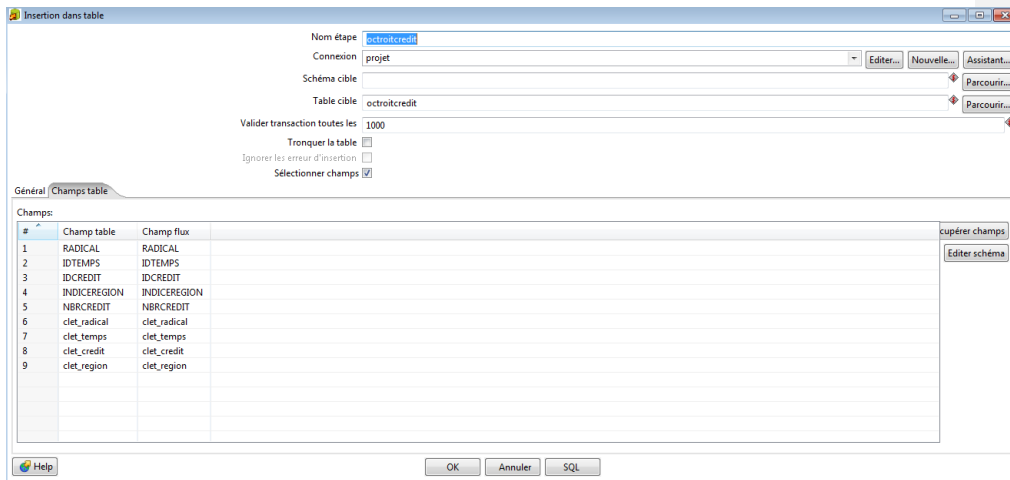


Figure III. 8 Alimentation de la table octroie crédit

Une fois que notre entrepôt de données a bien été alimenté ci-dessous les figures montrent les tables qui ont été créées et alimentées :

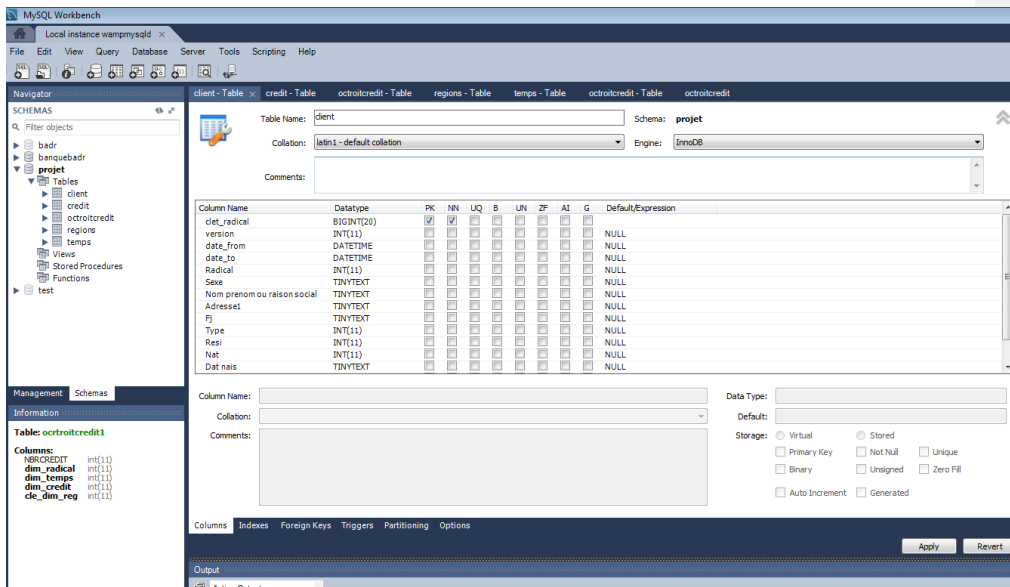


Figure III. 9 Table Client

det_radical	version	date_from	date_to	Radical	Sexe	Nom prenom ou raison social	Adresse1	Fj	Type	Resi	Nat	Dat nais	Dat ouv
0	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
1	1	1900-01-01 00:00:00	2199-12-31 23:59:59	S13000503	0	BOUCETTA ABDES	DNE LISMARIA A SAF SAF	PP	1	1	130	1901-01-01	1978-12-1
2	1	1900-01-01 00:00:00	2199-12-31 23:59:59	S13000580	0	KORTI MED SALIH	SIEGE COLS ABOU-TACFINE	PP	1	1	130	1949-09-11	1978-12-1
3	1	1900-01-01 00:00:00	2199-12-31 23:59:59	S13000610	0	BENSABEUR DJILALI	HAI EL NEDJMA N 107 IMMAMA	PP	1	1	130	1943-05-22	1978-12-1
4	1	1900-01-01 00:00:00	2199-12-31 23:59:59	S13000804	0	BOUTOUS SLIMANE O MOHAMED	ECOLE REGIONALE D AGRICULTURE	PP	1	1	130	1950-09-09	1978-12-1
5	1	1900-01-01 00:00:00	2199-12-31 23:59:59	S13000842	0	BELLATRECHE MOHAMMED	ROUTE DE SEBOU	PP	1	1	130	1950-09-01	1978-12-1
6	1	1900-01-01 00:00:00	2199-12-31 23:59:59	S13000875	0	BOUCHENTOUF BENATTOU	49 BD MED V	PP	1	1	130	1948-07-15	1978-12-1
7	1	1900-01-01 00:00:00	2199-12-31 23:59:59	S13000949	0	COOP DE CONSOMMATION CCLS	BP 62	PP	1	1	130	1978-01-08	1979-03-2
8	1	1900-01-01 00:00:00	2199-12-31 23:59:59	S13001002	0	BENABBOU ABDELHAMID	BENI MESTER	PM	1	1	130	1954-07-30	1978-12-1
9	1	1900-01-01 00:00:00	2199-12-31 23:59:59	S13001025	0	C C L S DE TLEMCEIN	ABOU TACHEFINE	PM	25	1	130	1982-01-01	1993-06-2
10	1	1900-01-01 00:00:00	2199-12-31 23:59:59	S13001066	0	MEDIANI OKACHA	11 RUE KAHDA TANI MED KEBIR	PP	1	1	130	1947-02-16	1979-03-0
11	1	1900-01-01 00:00:00	2199-12-31 23:59:59	S13001120	0	ZIDANE LABDELLI	ROUTE SIDI BEL ABDES	PP	1	1	130	1952-07-14	1991-11-1
12	1	1900-01-01 00:00:00	2199-12-31 23:59:59	S13001140	0	LACHACHI ABDERRAHIM	TERRAIN SEKKAL 23,6TS BT A N05	PP	1	1	130	1955-10-08	1979-09-0
13	1	1900-01-01 00:00:00	2199-12-31 23:59:59	S13001155	0	SLAM TLEMCEIN CERCLE JEUNES ALG	73 RUE DR DAMERDJI TLEMCEIN	PP	2	1	130	1989-09-18	1979-10-0

Figure III. 10 Données de la table client

Column Name	Datatype	PK	NN	UQ	B	LN	ZF	AI	G	Default/Expression
RADICAL	BIGINT(20)									NULL
IDTEMPS	INT(11)									NULL
INDICEREGION	BIGINT(20)									NULL
NBRCREDIT	BIGINT(20)									NULL
clet_radical	INT(11)									NULL
clet_temps	INT(11)									NULL
clet_credit	INT(11)									NULL
clet_region	INT(11)									NULL

Figure III. 11 Table octroie crédit

MySQL Workbench

Local instance wampmysqld

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

badr

banquebadr

projet

Tables

client

credit

octroicredit

regions

temps

Views

Stored Procedure

Functions

test

Management

Information

Table: octroicredit1

Columns:

NBR_CREDIT int(11)

dim_radical int(11)

dim_temps int(11)

dim_credit int(11)

cle_dim_reg int(11)

Query: SELECT * FROM projet.octroicredit;

Result Grid

RADICAL	IDTEMPS	IDCREDIT	INDICEREGION	NBR_CREDIT	det_radical	det_temps	det_credit	det_region
515000008	20020107	1	515	1	1000	11695	1	2
515014802	19781225	2	515	1	1001	3281	2	2
515014878	19781225	3	515	1	1002	3281	3	2
515014887	19781225	4	515	1	1003	3281	4	2
515015124	19791112	5	515	1	1004	3603	5	2
515015194	19800305	6	515	1	1005	3717	6	2
515015196	19800326	7	515	1	1006	3738	7	2
515015207	19800517	8	515	1	1007	3790	8	2
515015229	19800907	9	515	1	1008	3903	9	2
515015238	19801020	10	515	1	1009	3946	10	2
515015239	19801103	11	515	1	1010	3960	11	2
515015241	19801103	12	515	1	1011	3960	12	2
515015252	19801103	13	515	1	1012	3960	13	2
515015312	19810505	14	515	1	1013	4143	14	2
515015368	19810909	15	515	1	1014	4270	15	2
515015370	19810909	16	515	1	1015	4270	16	2

octroicredit 1 x

Output

Action Output

Figure III. 12 Données de la table octroie crédit

III.3.2 Création du cube

La création d'un cube au sens Mondrian est tout simplement la rédaction d'un fichier XML. Ce fichier permet de lier les informations du cube que nous souhaitons faire apparaître et la base de données.

Pour faciliter la création de ce fichier, nous allons utiliser l'outil Schéma Workbench qui offre une interface graphique pour effectuer cette tâche.

Après avoir configuré dans les « Préférences » la base de données sur laquelle nous souhaitons nous baser, nous pouvons commencer à créer notre cube.

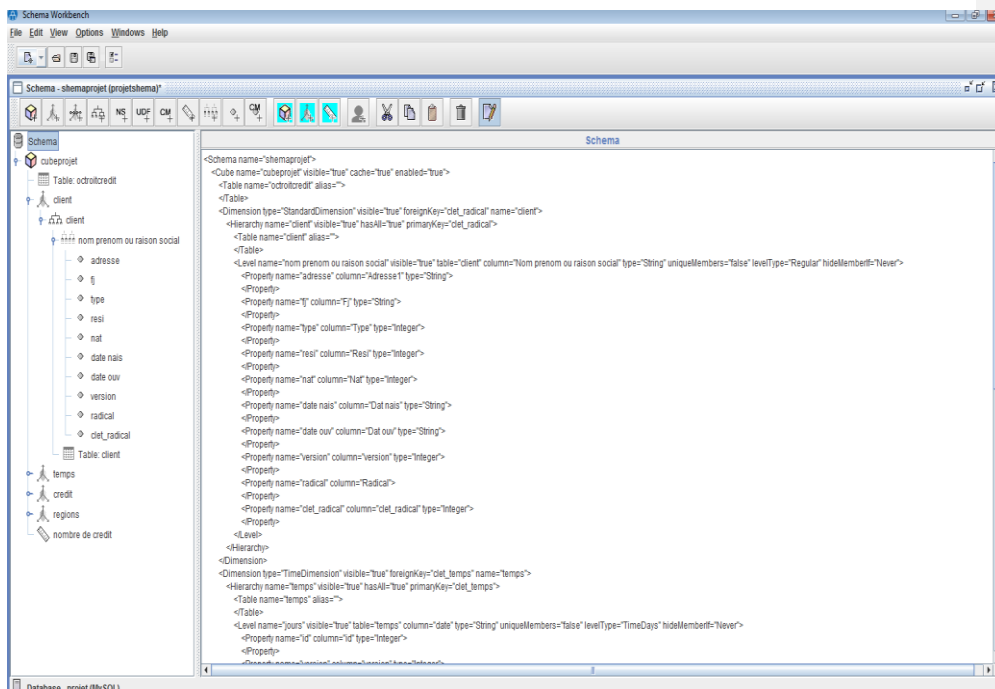


Figure III. 13Fichier XML

La figure ci-dessus montre que pour la création de notre cube nous avons eu besoin :

- De la table de fait « octroie crédit »
- Des dimensions « Client » « Crédit » «temps » « régions»
- De hiérarchies dimension et property pour chaque dimension
- De mesures dans notre cas « le nombre de crédit »

Une fois le cube créé nous procédons à sa publication dans le serveur Pentaho Mondrian en lui donnant l'URL de la console administration de Pentaho, le nom d'utilisateur et le mot de passe ensuite on nomme la source de donnée JNDI dans notre cas « projet ».

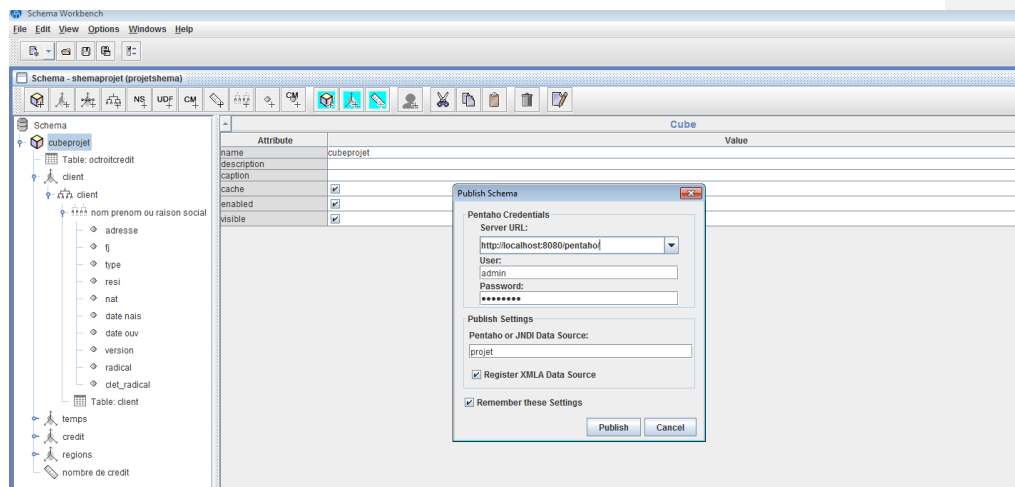


Figure III. 14 Publication du cube

III.3.3 Exploitation du cube

On crée dans un premier temps la connexion à notre base de données (projet). Pour cela, nous démarrons le BI Server de Pentaho mais surtout la console d'administration : se rendant dans les dossiers *bi-server* on lance le fichier de commande de Windows start-pentaho.bat.

On démarre ensuite dans notre navigateur la page local host:8080/Pentaho pour lancer la console d'administration, la page ci-dessous apparaîtra l'utilisateur *Admin* et le mot de passe par défaut *password*

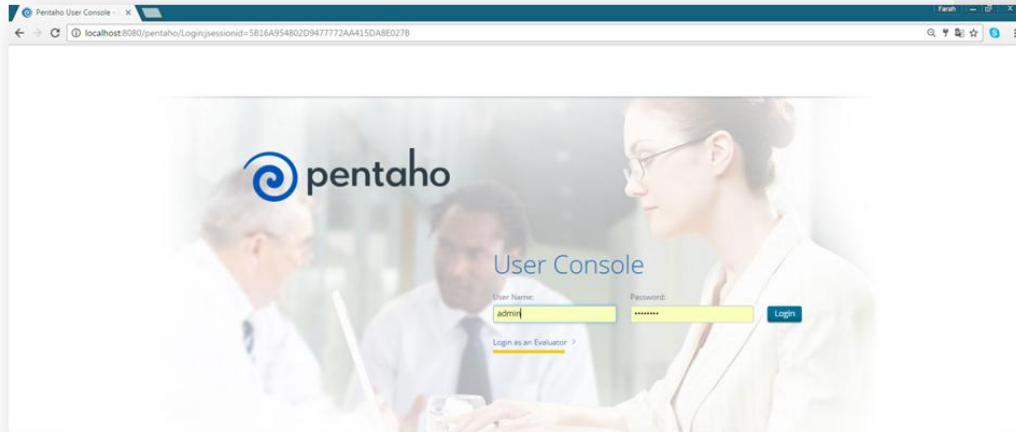


Figure III. 15 Authentification au serveur BI

Après être authentifié, l'interface principale de pentaho apparait comme suit :



Figure III. 16 Interface principale de pentaho

Afin d'utiliser et afficher nos données dans la plate - forme BI, nous allons procéder à la définition d'une source de données JDBC qui nécessite le même nom qui a déjà été défini précédemment lors de la publication du cube, ensuite nous procédons à la connexion à notre base de données de la manière suivante.

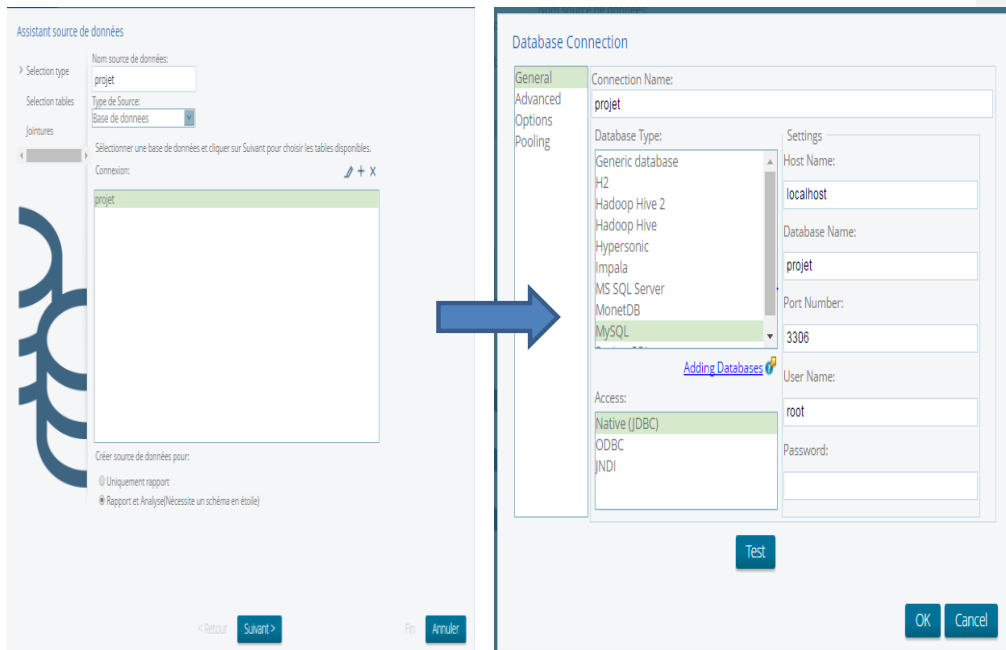


Figure III. 17 Création de la source de donnée et la connexion à la BDD

Une fois la connexion établie, nous procédons à deux étapes pour sélectionner les tables de la base de données et définir les jointures entre elles.

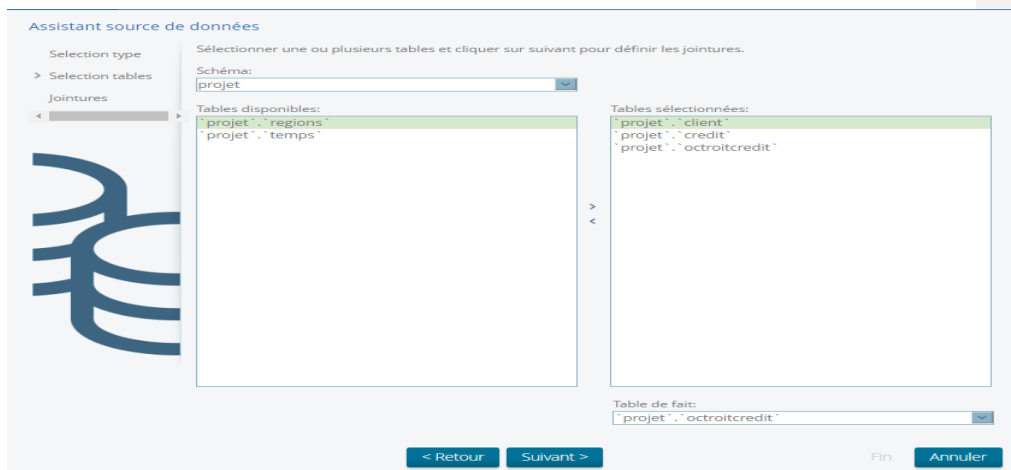


Figure III. 18 Sélection des tables

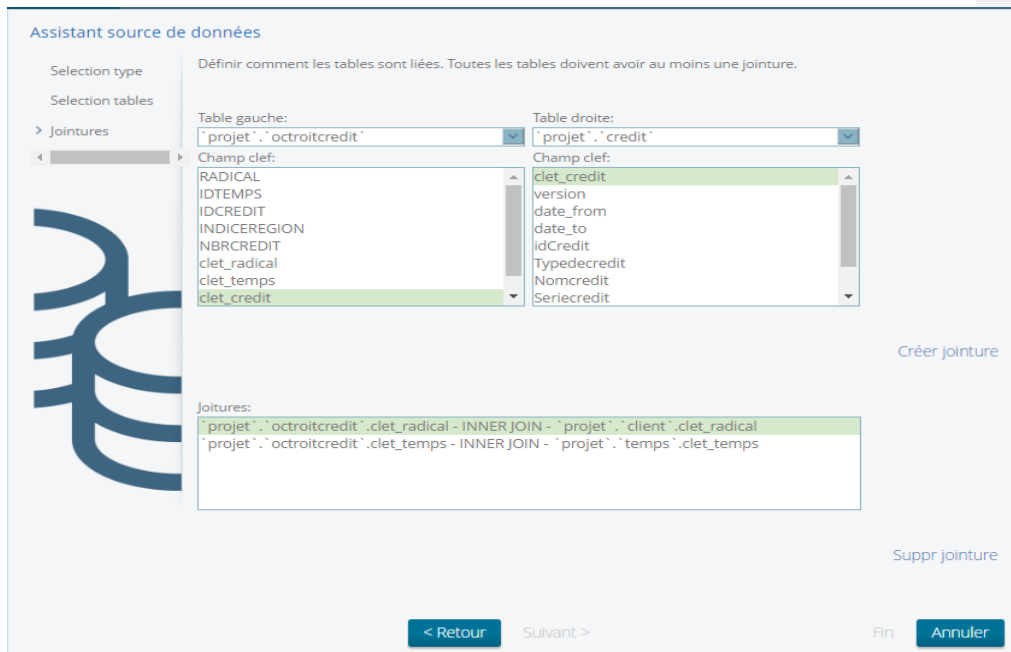


Figure III. 19 Définition des jointures

JPivot :

Afin d'obtenir une information d'une manière simple pour un utilisateur non informaticien a partir de notre cube nous avons utilisé l'outil graphique Jpivot, en effet il permet de transcrire des actions graphiques en requête MDX.

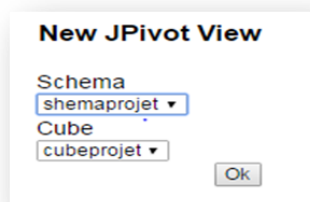


Figure III. 20 Ouverture d'une JPivotview

L'outil JPivot présente le cube sous forme de tableau croisé. Il permet d'insérer en abscisse ou en ordonné les différentes dimensions afin d'étudier les mesures du cube.

Voici un exemple d'une requête qui a été appliqué sur notre cube, nous allons ressortir le nombre de crédit octroyé par régions sur les différents types de crédits

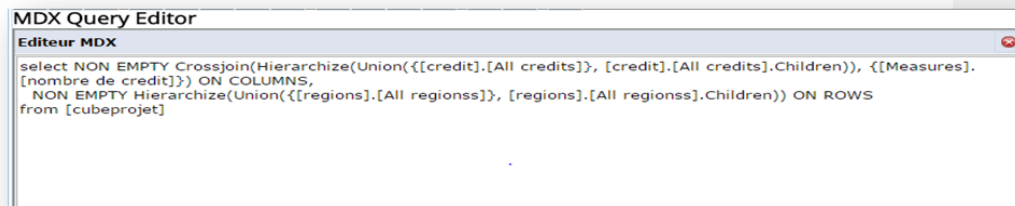


Figure III. 21 Requête MDX

regions	AVANCEE NEE CONST. DELEG. MARCHE	AVANCEE D'EXPLOITATION GARANTIE	AVANCE GARANTIE SECT. PUBL.	AVANCE SUR D.A.T. DEPOSITS A TERME	AVANCE SUR FACTURE SECT. PROVE	AVANCE SUR FACTURES MARCHANDISES	AVANCE SUR MARCHANDISES PUBL.	AVANCE SUR TITRES	AVANCEBONS DE CAISSE BADR
All regions	5 966	30	30	29	30	30	30	29	30
GHACHOUE	971	5	5	4	5	5	5	4	5
HENNAIA	999	5	5	5	5	5	5	5	5
REKICHI	999	5	5	5	5	5	5	5	5
SEDOU	999	5	5	5	5	5	5	5	5
TLIKICHI	999	5	5	5	5	5	5	5	5
TLIKICHI	999	5	5	5	5	5	5	5	5

Figure III. 22 Résultat de la requête MDX

III.3.4 Teste du tableau de bords

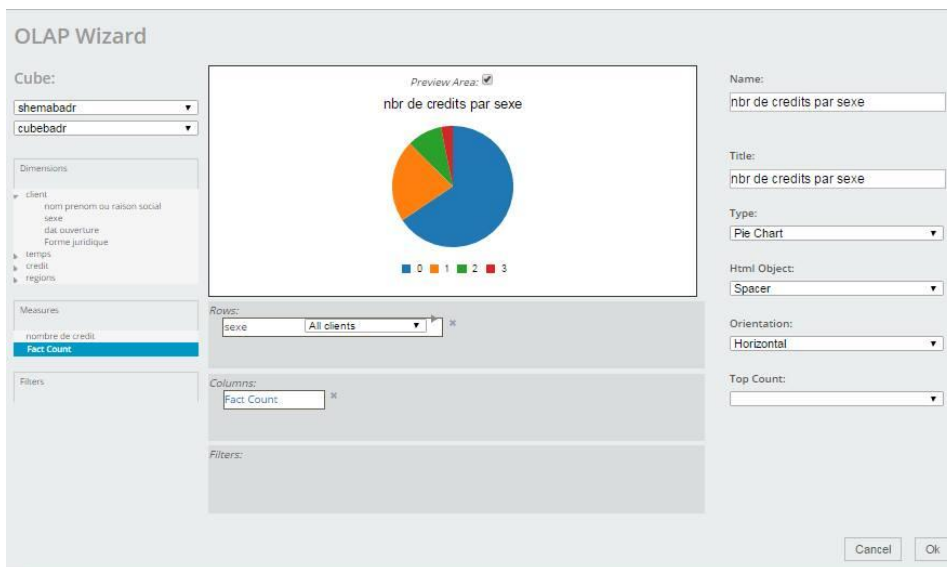


Figure III. 23 Teste pour réalisation d'un tableau de bord

III.4 Conclusion

Dans ce dernier chapitre, nous avons détaillé toutes les étapes liées à la construction du projet BI en y joignant les captures d'écran de l'entrepôt bancaire, tout en définissant les outils open source utilisés qui ont contribué à sa réalisation.

Conclusion générale et perspectives

L'informatique décisionnelle est un domaine en pleine évolution pour l'aide à la prise de décisions aux seins des entreprises. Elle est devenue un besoin capital, car la simple logique de produire pour répondre à une demande, ne suffit plus pour pérenniser l'activité de celle-ci, de ce fait sont apparus les entrepôts de données.

A travers ce mémoire, nous avons exposé les étapes de mise en place et de conception d'un entrepôt de données, à l'aide d'outils open source qui facilitera la prise de décisions aux seins de la banque BADR et permettre aux décideurs tout comme aux utilisateurs de sélectionner et filtrer des données utiles à la prise de décision.

Dans le premier chapitre, nous avons présenté des notions et des concepts sur les entrepôts de données, décrivant la méthode de modélisation existante d'un entrepôt de donnée et les différentes stratégies d'implémentation.

Dans le second chapitre, nous avons exposé notre travail, la modélisation de l'entrepôt de donnée bancaire utilisé, notre schéma en étoile, ainsi que la méthode de modélisation.

Enfin, dans le troisième chapitre, nous avons présenté les outils utilisés pour la réalisation de notre entrepôt de données, et l'explication des étapes de modélisations en image sur ces outils

Dans les perspectives à venir, nous projetons de :

- Basculer vers d'autres éditeurs plus performants exemple : éditeur Saiku.
- Rajouter d'autres requêtes MDX.
- Etendre l'entrepôt en ajoutant d'autres magasins de données, exemple « l'analyse sur le nombre d'ouverture des comptes ».
- Exploiter les tableaux de bords.
- Tester ce travail a la BANK BADR avec des données plus importantes.

Références Bibliographique

- [1] Aftab Ahmed, Data & Analytics : FORIS Business Intelligence for Telecom, slideshare, article 24 mars 2016.
- [2] LeMagIT, BI (informatique décisionnelle): <http://www.lemagit.fr/definition/BI-informatique-decisionnelle>, visité le 18 septembre 2016
- [3] Toby Dean, Gain the Competitive Edge with Business Intelligence Software & Analytics. Business 2 community, article July 25, 2015.
- [4] Piloter la performance : Data Warehouse, Entrepôt de données : <http://www.piloter.org/business-intelligence/datawarehouse.htm> , visité le 19 septembre 2016
- [5] Selma khouri, Modélisation conceptuelle à base ontologique d'un entrepôt de données. Ecole nationale Supérieure d'Informatique (E.S.I), mémoire, 2008/2009.
- [6] SernaEncinas Maria Trinidad. Entrepôts de données pour l'aide à la décision médicale: conception et expérimentation. Networking and Internet Architecture. Université Joseph-Fourier - Grenoble I, 2005.
- [7] Omar SENHAJI, Data WareHouse: Entrepôt de données. supinfo international university. article 31/10/2015
- [8] Marlyse Dieungang, Khaoula Ghilani, Datawarehouse: Cubes OLAP. Université libre de bruxelles
- [9] Bernard ESPINASSE, Entrepôts de données : Introduction au langage MDX (Multi-Dimensionale Extensions) pour l'OLAP, Ecole Polytechnique, support Décembre 2015.
- [10] Bernard ESPINASSE, Entrepôts de données : Systèmes ROLAP, MOLAP et HOLAP, Ecole Polytechnique Universitaire de Marseille, support Décembre 2015
- [11] SGBD et Datawarehouse L'Institut d'électronique et d'informatique Gaspard-Monge (IGM) . Université Paris-Est Marne-la-Vallée , exposé 2005.
- [12] R. Godin, C. Desrosiers, Les entrepôts de données et l'analyse de données. Département de génie logiciel et des TI Tableau de bord, ETS - Hiver 2011

[13]Cécile Favre, FadilaBentayeb, Omar Boussaid, Jérôme Darmont,Gérald Gavin, NouriaHarbi, Nadia Kabachi, Sabine Loudcher, Les entrepôts de données pour les nuls. . . ou pas !. Université de Lyon, article

[14]Alexis Rousset, navigation au sein d'un cube OLAP. Blog netapsys.article.28 mai 2009

[15]FILALI ABDERRAHMANE, KEDJNANE SOFIANE. Conception et réalisation d'un Data Warehouse pour la mise en place d'un système décisionnel. Ecole nationale Supérieure d'Informatique (E.S.I), mémoire, 2009/2010.

[16]<https://www.google.dz/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=&url=http%3A%2F%2Fwww.lejournaldesentreprises.com%2Feditions%2F76%2Fdossiers%2Fgerer%2Ftableaux-de-bord-quels-sont-les-indicateurs-incontournables-07-10-2011-133811.php&psig=AFQjCNEAf2Sujh3VIyLNBvLzRlKH-oEgEw&ust=1480787830337060> , Visité le 29/11/2016

[17] MySQL Workbench, Guide de modélisation des données pour les développeurs et les DBA, article, oracle 2010.

[18] Sherman Wood, Mondrian Schema Workbench, JasperSoft, article, 2007

[19] <http://www.osbi.fr/cubes-olap-avec-mondrian/>visité le 12/11/2016

[20] Sylvain, présentation et installation de pentaho, Develloper.com: club des développeurs et IT pro, tutoriel, mars 2012

[21] <http://www.wolfgroup.co.uk/tag/business-intelligence/> visité le 24/10/2016

Liste des figures

Figure I. 1 Modèle du processus de prise de décision [1].....	3
Figure I. 2 business intelligence [21].....	4
Figure I. 3 Entrepôt de données [5]	5
Figure I. 4 Caractéristiques d'un data warehouse	5
Figure I. 5 Schéma en étoile	8
Figure I. 6 Schéma en flocon de neige	9
Figure I. 7 Schéma en constellation	10
Figure I. 8 Schéma en cube	11
Figure I. 9 Tableau de bord [16].....	19
Figure II. 1 Logo de la banque BADR.....	21
Figure II. 2 Modèle en étoile.....	22
Figure II. 3 Exemple fichier CSV Client	23
Figure II. 4 Exemple de table dans la base de données.....	24
Figure III. 1 Architecture technique de l'entrepôt bancaire	29
Figure III. 2 Extraction, transformation et chargement de la dimension client.....	33
Figure III. 3 Extraction des données à partir d'un fichier CSV	34
Figure III. 4 Configuration générale	34
Figure III. 5 Configuration du champ clé	35
Figure III. 6 Configuration champs sujet aux changements	35
Figure III. 7 Création de la table des faits octroie crédit.....	36
Figure III. 8 Alimentation de la table octroie crédit	37
Figure III. 9 Table Client	37
Figure III. 10 Données de la table client	38
Figure III. 11 Table octroie crédit	38
Figure III. 12 Données de la table octroie crédit	39
Figure III. 13 Fichier XML	40
Figure III. 14 Publication du cube.....	41
Figure III. 15 Authentification au serveur BI.....	42
Figure III. 16 Interface principale de pentaho	42
Figure III. 17 Création de la source de donnée et la connexion à la BDD	43
Figure III. 18 Sélection des tables	43
Figure III. 19 Définition des jointures	44
Figure III. 20 Ouverture d'une JPivotview.....	44
Figure III. 21 Requête MDX	45
Figure III. 22 Résultat de la requête MDX	45
Figure III. 23 Teste pour réalisation d'un tableau de bord	45

Liste des tableaux

Tableau I. 1 Comparaison entre OLTP et OLAP	15
Tableau I. 2 Comparaison entre SQL et MDX.....	17

Résumé

L'entreposage de données est une phase du processus décisionnel qui supporte efficacement le processus OLAP, il est né dans les entreprises pour l'aide à la prise de décision. Ainsi, les principaux utilisateurs de ces technologies font partie intégrante de l'entreprise, Nous pouvons citer les secteurs qui peuvent tirer profit des outils d'aide à la décision comme les banques et les assurances, ainsi que ceux de l'automobile et des institutions médicales.

La réalisation de l'entrepôt de données au sein de la BADR BANK, facilitera la prise de décision, et donnera une vision plus claire sur la situation des crédits octroyés, en donnant la possibilité aux décideurs de connaître la rentabilité de chaque agence donc la rentabilité par région, de savoir lequel des crédits est à financer le plus ou le moins.

ملخص

طبق في الشركات OLAP تخزين البيانات هو مرحلة من مراحل عملية اتخاذ القرار التي تدعم بشكل فعال عملية للمساعدة في اتخاذ القرارات. وهكذا، فإن المستخدمين الرئيسيين لهذه التقنيات هو جزء لا يتجزأ من هذا الأخير، يمكن أن نذكر القطاعات التي يمكن أن تستفيد من أدوات دعم القرار مثل البنوك والتأمينات السيارات و المؤسسات الطبية. تحقيق مستودع البيانات في بنك بدر، يسهل اتخاذ القرارات ويعطي صورة أوضح عن حالة القروض الممنوحة، بإعطاء الفرصة لصانعي القرار لمعرفة ربح كل وكالة بالتالي ربح كل منطقة، لمعرفة أي من القروض صالحة لتمويل أكثر أو أقل.

Abstract

Data warehousing is a phase of process decision-making that effectively supports the OLAP process was born in business for help decision-making. Thus, the main users of these technologies are an integral part of the latter. We can cite sectors that can benefit from decision-making tools such as banks and insurance, as well as those of the automobile and medical institutions. The realization of the data warehouse within BADR BANK will facilitate decision-making and give a clearer picture of the situation of the granted credits, giving decision-makers the opportunity to know the profitability of each agency thus the profitability by region, to know which of the credits is to be financed more or less.